



(11) **EP 2 401 682 B1**

(12) **EUROPEAN PATENT SPECIFICATION**

(45) Date of publication and mention
of the grant of the patent:
03.05.2017 Bulletin 2017/18

(51) Int Cl.:
G06F 15/173 ^(2006.01) **H04L 12/43** ^(2006.01)
H04L 12/42 ^(2006.01) **H04L 12/801** ^(2013.01)
H04L 12/721 ^(2013.01)

(21) Application number: **10709293.4**

(86) International application number:
PCT/JP2010/052805

(22) Date of filing: **17.02.2010**

(87) International publication number:
WO 2010/098330 (02.09.2010 Gazette 2010/35)

(54) **INFORMATION PROCESSING APPARATUS AND OPERATION METHOD THEREOF, COMPUTER PROGRAM, AND STORAGE MEDIUM**

INFORMATIONSVERRARBEITUNGSVORRICHTUNG UND BETRIEBSVERFAHREN DAFÜR,
COMPUTERPROGRAMM UND SPEICHERMEDIUM

APPAREIL DE TRAITEMENT DE L'INFORMATION ET PROCÉDÉ D'EXPLOITATION DUDIT,
PROGRAMME INFORMATIQUE ET SUPPORT DE STOCKAGE

(84) Designated Contracting States:
**AT BE BG CH CY CZ DE DK EE ES FI FR GB GR
HR HU IE IS IT LI LT LU LV MC MK MT NL NO PL
PT RO SE SI SK SM TR**

(74) Representative: **Hitching, Peter Matthew et al**
Canon Europe Ltd
European Patent Department
3 The Square
Stockley Park
Uxbridge Middlesex UB11 1ET (GB)

(30) Priority: **25.02.2009 JP 2009043148**

(43) Date of publication of application:
04.01.2012 Bulletin 2012/01

(56) References cited:
EP-A1- 2 336 897 JP-A- 2010 217 959
JP-A- 2010 244 512 JP-A- 2010 245 953
US-A- 4 809 267 US-A1- 2002 191 601
US-B2- 7 414 984

(60) Divisional application:
16150623.3 / 3 037 978

(73) Proprietor: **Canon Kabushiki Kaisha**
Tokyo 146-8501 (JP)

• **ANONYMOUS: 'Bubble (computing)', [Online] 04
April 2008, XP055208408 Retrieved from the
Internet:
<URL:https://en.wikipedia.org/w/index.php?title=Bubble_(computing)&oldid=203253475>
[retrieved on 2015-08-18]**

(72) Inventors:
• **TAKASAKA, Michiaki**
Tokyo 146-8501 (JP)
• **ISHIKAWA, Hisashi**
Tokyo 146-8501 (JP)

Note: Within nine months of the publication of the mention of the grant of the European patent in the European Patent Bulletin, any person may give notice to the European Patent Office of opposition to that patent, in accordance with the Implementing Regulations. Notice of opposition shall not be deemed to have been filed until the opposition fee has been paid. (Art. 99(1) European Patent Convention).

EP 2 401 682 B1

Description

TECHNICAL FIELD

[0001] The present invention relates to an information processing apparatus and operation method thereof, a computer program, and a storage medium.

BACKGROUND ART

[0002] Various ring-shaped communication networks (circuit networks) have been proposed and have prevailed. For example, the LAN (Local Area Network) specification includes a so-called token ring (see ISO/IEC 8802-5: 1998).

[0003] Data transfer in a token ring system will be briefly described below. A node that wants to transmit data acquires a token called a free token, which is not occupied by any node, and is going around a ring bus. The node that acquired the free token copies a destination identifier and processing data to the token, and outputs that token onto the ring bus to transmit it to a destination module. As described above, in the token ring system, a frame called a token is transferred in turn from a node to another node, and then to the next node, which are allocated on the ring bus, thus transmitting that frame to a target terminal.

[0004] On the other hand, when a receiving node receives a token which has a destination as the self node, it copies processing data and enters the received token, which is set with a reception completion flag indicating completion of reception, again onto the ring bus. This entry is done to inform the transmission source node of successful data transfer, and the transmission source node re-transmits the identical token if the token which is not set with the reception completion flag is returned. In this way, the receiving node sets the reception completion flag in the token, and returns that token to the transmission source node, thus implementing data transfer.

[0005] US 2002/0191601 A1 discusses an on-chip communication architecture and method. A chip is described that incorporates a distributed bus architecture which enable packets of data to be communicated between multiple source/destination modes. The chip can be designed such that each source/destination node monitors a connecting path (e.g. bus) for receipt of a packet of data. Upon receiving a packet of data at one of the source/destination nodes, that source/destination node then determines whether the packet of data is addressed to that source/destination node. If the packet of data is addressed to that source/destination node, then that node takes the data from the packet of data. Otherwise, if the packet of data is not addressed to that source/destination node, then that node transmits the packet of data on the connecting path to a neighboring source/destination node. The source/destination nodes can also insert data into a packet of data addressed to

another source/destination node.

DISCLOSURE OF INVENTION

[0006] When an information processing system, in which a plurality of modules that execute data processing are connected to a ring bus, adopts the aforementioned token ring bus system, a token has to be returned to a transmission source irrespective of a success or failure of reception. That is, even when the reception has succeeded, no one can copy data to the token until the transmission source releases the returned token, resulting in inefficiency.

[0007] The present invention allows efficient packet transfer in an information processing system, in which a plurality of modules that execute data processing are connected to a ring bus.

[0008] One aspect of embodiments of the present invention relates to an information processing apparatus including a plurality of modules, which are connected to a ring bus, and each of which is configured to receive a packet that stores data from one neighboring module, and to transmit the packet to the other neighboring module after predetermined processing, characterized in that, each module comprises, identification means for identifying a packet received from another module to determine whether or not the received packet is a packet to be processed by a self-module, reception means for, when the identification means determines that the received packet is a packet to be processed by the self-module, extracting data to be processed from the packet, processing means for processing the extracted data, modification means for, when the processing means processes the data, setting a first flag indicating validity of data in the received packet to have a value indicating that the data is invalid, and transmission means for transmitting the packet to the other module; wherein when said processing means is not ready to process the data, said modification means modifies a value of a second flag in the received packet from an initial value.

[0009] Further features of the present invention will become apparent from the following description of exemplary embodiments (with reference to the attached drawings).

BRIEF DESCRIPTION OF DRAWINGS

[0010]

Fig. 1 is a block diagram showing an example of the arrangement of a data processing system;
Fig. 2 is a block diagram showing an example of the arrangement of a data processor 104;
Fig. 3 is a block diagram showing an example of the arrangement of a module;
Fig. 4 is a view showing an example of the format of a packet;
Fig. 5 is a flowchart showing an example of process-

ing in a processing data transmission unit 303 and reception unit 311;

Fig. 6 is a flowchart showing an example of processing in a processing data reception unit 305 and transmission unit 312;

Fig. 7 is a block diagram showing an example of the arrangement of a module; and

Fig. 8 is a flowchart showing an example of processing in a processing data transmission unit 303 and reception unit 311.

BEST MODE FOR CARRYING OUT THE INVENTION

[0011] Embodiments of the present invention will be described hereinafter. This embodiment relates to a data processing system in which a plurality of modules that execute data processing are connected to a ring bus. Of course, the embodiments to be described hereinafter provide disclosures for facilitating implementation by those who are skilled in the art of the present invention, and are just some embodiments included in the technical scope of the present invention settled by the scope of the claims. Therefore, it is obvious for those who are skilled in the art that even embodiments that are not directly described in the specification of the present invention are incorporated in the technical scope of the present invention as long as they have common technical ideas.

[0012] Note that a plurality of embodiments will be described for convenience sake. However, as is easily understood for those who are skilled in the art, not only these embodiments can form independent inventions, but a plurality of embodiments can also be combined as needed to form the invention.

[0013] The arrangement of an information processing system according to an embodiment corresponding to one aspect of the present invention will be described below with reference to Fig. 1. A CPU 101 is a control unit having a function of controlling the overall operations of a data processing system. A RAM 102 is a readable/writable memory which stores input data to be processed by the CPU 101, output data after the processing, preset parameter data to a data processor 104, and the like. A ROM 103 is a readable memory that can hold the processing sequence, constants such as preset parameters, and the like of the CPU 101. The data processor 104 includes an arrangement and modules disclosed by the present invention. Respective modules are connected to a ring bus. A programmable custom IC chip may implement the data processor 104. The chip includes, for example, an ASIC (Application Specific Integrated Circuit) or FPGA (Field Programmable Gate Array).

[0014] The arrangement of the data processor 104 will be described below with reference to Fig. 2. The data processor 104 includes a plurality of modules 201 to 204. A ring bus 205 is used to transfer packets among the modules. The modules 201 to 204 are connected to the ring bus 205. Each module is configured to receive a data packet from one neighboring module, and to transmit the

data packet to the other neighboring module after predetermined processing. A packet exchanged by the modules moves in one direction on the ring bus. In the following description, a ring-shaped bus on which data (or packet) goes around in a single direction will be simply referred to as a ring bus.

[0015] An input/output buffer 206 is used to input/output data. Each of the modules 201 to 204 of this embodiment has data processor identification information, which is used to identify the self-module, and is described as "ID" in Fig. 2 for the sake of simplicity. In the following description of the embodiments of the present invention, that information is also described as "ID". Note that the module 201 with ID = 1 is connected to the input/output buffer 206 to manage input and output data.

[0016] The arrangement of the respective modules 201 to 204 in the data processor will be described below with reference to Fig. 3. Referring to Fig. 3, a packet identification unit 301 checks whether or not to fetch a packet coming from another module. When the packet identification unit 301 receives a packet holding a transmission source ID that matches a self waiting ID, it outputs the packet to a packet reception unit 302.

[0017] The packet reception unit 302 executes packet reception processing. A processing data transmission unit 303 transmits processing data to a processing unit 304. The processing unit 304 executes actual data processing. A processing data reception unit 305 receives processed data from the processing unit 304. A packet generation unit 306 generates a packet based on the processed data received from the processing unit 304 and an output from a packet modification unit 308.

[0018] A packet transmission unit 307 outputs a packet onto the ring bus. The packet modification unit 308 modifies values of a data valid flag 401 and stall flag 404 of a packet, which will be described later with reference to Fig. 4, in accordance with an instruction from the packet reception unit 302. A selector 309 selects one of the packets from the packet transmission unit 307 and that from the packet modification unit 308 based on an instruction from the packet transmission unit 307. The selector 309 selects the packet output from the packet modification unit 308 and outputs that packet if there is no instruction from the packet transmission unit 307. A buffer 310 is used to transfer a packet one after another.

[0019] In the above arrangement, each of the modules 201, 202, 203, and 204 includes the processing unit 304 and a communication unit 313. The communication unit 313 includes the processing data transmission unit 303, the processing data reception unit 305, the selector 309, the buffer 310, a reception unit 311, and a transmission unit 312. Furthermore, the reception unit 311 includes the packet identification unit 301, packet reception unit 302, and packet modification unit 308. Also, the transmission unit 312 includes the packet generation unit 306 and packet transmission unit 307.

[0020] In the following description, this embodiment will explain a case in which the processing unit 304 out-

puts one output data by processing one input data. In this case, assume that another input data cannot be input from when input data is input to the processing unit 304 until output data is obtained. For this reason, the packet reception unit 302 has to suspend reception of a packet according to the status of the processing unit 304.

[0021] Fig. 4 shows the format of a packet required to exchange data among the modules. The data valid flag 401 is first flag information used to check validity of data in a packet. For example, if the data valid flag 401 is "1", data is valid; if it is "0", data is invalid.

[0022] A transmission source ID 402 is an identifier of a transmission source that stores an identifier of the module that transmitted a packet. A data storage field 403 stores a processing data main body. The stall flag 404 is second flag information, which is set to be "1" when the module suspends packet processing. That is, if the stall flag 404 is "1", it indicates that the packet processing is suspended. Note that the processing is suspended, for example, when a certain module receives data but it cannot process the data since the processing unit 304 is busy. The stall flag 404 is set to be "0" as an initial value. That is, when the stall flag has the initial value, this means that a module on the ring bus, which is to receive that packet, has not processed the corresponding packet yet.

[0023] Each module can set a waiting packet ID. The packet identification unit 301 fetches a packet when the waiting packet ID matches the transmission source ID 402 of the packet that flows through the ring bus. For example, a case will be examined below wherein a data path is to be configured in the order of module 1, module 4, module 2, module 3, and module 1. In this case, the waiting packet ID of module 1 is set to be "3", that of module 2 is set to be "4", that of module 3 is set to be "2", and that of module 4 is set to be "1". By respectively setting the waiting packet IDs of the modules in this way, the data path can be formed. Assume that the waiting packet ID is set in advance for each module, and its information is stored in the packet identification unit 301.

[0024] A data reception method in the communication unit 313 will be described below. For example, a case will be examined below wherein data transfer is executed from module 1 with ID = 1 to module 3 with ID = 3 in Fig. 2. At this time, the packet identification unit 301 of module 3 acquires a packet in which the data valid flag 401 is valid, and compares the transmission source ID 402 of that packet with the self waiting packet ID. If these IDs are equal to each other, the packet identification unit 301 transmits that packet to the packet reception unit 302.

[0025] The processing data transmission unit 303 determines whether or not the processing unit 304 is ready to execute processing. If the processing data transmission unit 303 determines that the processing unit 304 is ready, the packet reception unit 302 extracts data to be processed from the acquired packet, and transmits it to the processing data transmission unit 303. In this case, since that packet already has no use, the packet reception unit 302 instructs the packet modification unit 308 to

set the data valid flag 401 of the packet to be invalid (0).

[0026] On the other hand, if the processing data transmission unit 303 determines that the processing unit 304 is busy, the packet reception unit 302 instructs the packet modification unit 308 to set the stall flag 404 of the packet to be "1". The packet modification unit 308 modifies the data valid flag 401 or stall flag 404 according to an instruction from the packet reception unit 302, and transmits the packet to the selector 309. The selector 309 selects one of the packet input from the packet modification unit 308 and that input from the packet transmission unit 307, and transmits it to the buffer 310.

[0027] The processing in the processing data transmission unit 303 and reception unit 311 will be described below with reference to Fig. 5. In step S501, the packet identification unit 301 acquires a packet from a module located at a previous stage. The packet identification unit 301 checks in step S502 if the data valid flag 401 of the packet is valid (1). If the data valid flag 401 is valid ("YES" in step S502), the process advances to step S503. On the other hand, if the data valid flag 401 is not valid ("NO" in step S502), this processing ends.

[0028] The packet identification unit 301 checks in step S503 if the transmission source ID 402 of the packet is equal to the waiting packet ID set in the module. If it is determined that the two IDs are equal to each other ("YES" in step S503), the process advances to step S504. On the other hand, if it is determined that the two IDs are not equal to each other ("NO" in step S503), this processing ends.

[0029] The processing data transmission unit 303 checks in step S504 if the processing unit 304 is ready to exchange data. If it is determined that the processing unit 304 is ready to exchange data ("YES" in step S504), the process advances to step S505. On the other hand, if it is determined that the processing unit 304 is not ready to receive data ("NO" in step S504), the process advances to step S506.

[0030] In step S505, the packet reception unit 302 extracts processing data from the packet, and transmits it to the processing data transmission unit 303, which passes the data to the processing unit 304. The process then advances to step S507. In step S506, the packet reception unit 302 instructs the packet modification unit 308 to set the stall flag 404 of the packet to be "1". Then, the stall flag 404 is modified from the initial value (0) to (1). In step S507, the packet reception unit 302 instructs the packet modification unit 308 to set the data valid flag 401 of the packet to be "0".

[0031] The data transmission sequence in the processing data reception unit 305 and transmission unit 312 will be described below. For example, a case will be examined below wherein data transfer is executed from module 1 with ID = 1 to module 3 with ID = 3 in Fig. 2. At this time, the packet generation unit 306 of module 1 acquires a packet in which the data valid flag 401 is invalid (0) from the packet modification unit 308.

[0032] Then, the packet generation unit 306 stores

transmission data obtained from the processing data reception unit 305 in the data storage field 403 of that packet, and stores "1" as the ID of module 1 in the transmission source ID 402. Also, the packet generation unit 306 sets the stall flag 404 to be the initial value (0), and sets the data valid flag 401 to be "1". Then, the packet generation unit 306 transmits the packet to the packet transmission unit 307. The packet transmission unit 307 outputs the packet acquired from the packet generation unit 306 to the selector 309. At this time, the packet transmission unit 307 simultaneously outputs a select signal to the selector 309 so as to select its output.

[0033] The packet generation unit 306 monitors the stall flag 404 of the packet. When the packet output by itself is returned without setting the stall flag to be "1", the data valid flag 401 is set to be "0" so as to avoid the packet having no receiving module from occupying the ring bus. Conversely, when the stall flag 404 is set to be "1", that packet is output intact onto the ring bus.

[0034] The processing in the processing data reception unit 305 and transmission unit 312 will be described below with reference to Fig. 6. Referring to Fig. 6, the packet generation unit 306 acquires a packet from the packet modification unit 308 of the reception unit 311 in step S601. The packet generation unit 306 checks in step S602 if the data valid flag 401 of the received packet is "0". If it is determined that the data valid flag 401 is not "0" ("NO" in step S602), the process advances to step S603. On the other hand, if it is determined that the data valid flag 401 is "0" ("YES" in step S602), the process advances to step S606.

[0035] The packet generation unit 306 checks in step S603 if the stall flag 404 of the packet is "1". If it is determined that the stall flag 404 is "1" ("YES" in step S603), the processing ends. In this case, the packet from the packet modification unit 308 is transmitted to the next module via the selector 309 and buffer 310. On the other hand, if it is determined that the stall flag 404 is not "1", i.e., it still has the initial value ("NO" in step S603), the process advances to step S604.

[0036] The packet generation unit 306 checks in step S604 if the transmission source ID 402 of the packet is equal to the ID of the self-module. If it is determined that the two IDs are equal to each other ("YES" in step S604), the process advances to step S605. In step S605, the packet generation unit 306 sets the data valid flag 401 of the packet to be "0", and the process then jumps to step S611. On the other hand, if it is determined that the two IDs are not equal to each other ("NO" in step S604), this processing ends. In this case as well, the packet from the packet modification unit 308 is transmitted to the next module via the selector 309 and buffer 310.

[0037] The packet generation unit 306 checks in step S606 if valid processed data is obtained from the processing unit 304. If the valid processed data is obtained ("YES" in step S606), the process advances to step S607. On the other hand, if no valid processed data is obtained ("NO" in step S606), this processing ends.

[0038] In step S607, the packet generation unit 306 copies the processed data acquired from the processing unit 304 to the data storage field 403 of the packet. In step S608, the packet generation unit 306 sets the data valid flag 401 of the packet to be "1". Furthermore, in step S609 the packet generation unit 306 sets the stall flag 404 of the packet to be the initial value (0). Moreover, in step S610 the packet generation unit 306 sets the self ID in the transmission source ID 402 of the packet. In step S611, the packet generation unit 306 instructs the selector 309 to preferentially select the input from the packet transmission unit 307.

[0039] As described above, according to this embodiment, when the receiving module can process data of a received packet, it invalidates the packet; when the receiving module cannot process data, it sets the stall flag 404 of the packet to be "1" and re-enters the packet onto the ring bus. The receiving module can manage a packet to be received so as to prevent a re-transmission request packet to the transmitting module from being output onto the ring bus. Since the transmitting module monitors the stall flag 404 of the output packet and invalidates the packet as needed, an unnecessary packet can be prevented from going around on the ring bus, thus enhancing the data transfer efficiency among the modules.

[0040] In this embodiment, the number of destination modules of the packet is one. However, the number of destinations is not limited to one, and the data path may branch. For example, this is the case when modules 3 and 4 wait for an output packet from module 2 in Fig. 2. That is, the data path after module 2 branches.

[0041] In the above embodiment, when data is successfully transferred to the processing unit 304, the received packet is invalidated. However, if the packet is similarly invalidated since module 3 successfully receives the packet, module 4 can no longer receive the packet.

[0042] Hence, each module according to an embodiment corresponding to another aspect of the present invention is characterized by further including a packet invalidate processing instruction register which can designate whether or not to permit invalidation of a packet when data is successfully transferred to the processing unit 304.

[0043] Fig. 7 is a block diagram showing an example of the arrangement of a module according to this embodiment. The module block diagram shown in Fig. 7 is basically the same as that shown in Fig. 3. However, unlike in Fig. 3, a reception unit 311 further includes an invalidation instruction register 701. In this embodiment, when this register preset value is "1", it is permitted to invalidate a received packet by setting the value of a data valid flag 401 at the time of successful reception by a processing unit 304. On the other hand, if the preset value is "0", it is not permitted to invalidate a received packet by setting the value of the data valid flag 401 at the time of successful reception by the processing unit 304. However, the setting specification of the packet invalidate process-

ing instruction register is not limited to the above.

[0044] The invalidation instruction register 701 is set as follows. When it is assumed that a data path does not branch, the packet invalidate processing instruction registers of all modules are set to be "1". In this case, at the time of successful reception, a packet is invalidated.

[0045] On the other hand, when a plurality of modules wait for an output packet from a certain module, a transmission source module which outputs a packet that the plurality of modules wait for is specified first, and the plurality of destination modules which wait for that packet are similarly specified.

[0046] Of the plurality of destination modules, a module which is allocated at a position farthest from the transmission source module along a ring bus is specified, and the invalidation instruction register 701 of this module is set to be "1". The invalidation instruction registers 701 of modules other than that module are set to be "0".

[0047] For example, when the data path branches at one position like in a case in which modules 3 and 4 wait for an output packet from module 2 in Fig. 2, the ring bus is traced from module 2 as the transmission source module in a forward direction. Of modules 3 and 4 that wait for a packet, module 4 is allocated at a position farthest from module 2. Hence, the invalidation instruction register 701 of module 3 is set to be "0", and that of module 4 is set to be "1".

[0048] The processing in a processing data transmission unit 303 and reception unit 311 in this embodiment will be described below with reference to Fig. 8. Referring to Fig. 8, a packet identification unit 301 acquires a packet from a module located at a previous stage in step S801. The packet identification unit 301 checks in step S802 if the data valid flag 401 of the packet is "1". If the data valid flag 401 is "1" ("YES" in step S802), the process advances to step S803. On the other hand, if the data valid flag 401 is not "1" ("NO" in step S802), this processing ends.

[0049] The packet identification unit 301 checks in step S803 if a transmission source ID 402 of the packet is equal to a preset waiting packet ID. If it is determined that the two IDs are equal to each other ("YES" in step S803), the process advances to step S804. On the other hand, if it is determined that the two IDs are not equal to each other ("NO" in step S803), this processing ends.

[0050] The processing data transmission unit 303 checks in step S804 if the processing unit 304 is ready to exchange data. If it is determined that the processing unit 304 is ready to exchange data ("YES" in step S804), the process advances to step S805. On the other hand, if it is determined that the processing unit 304 is not ready to exchange data ("NO" in step S804), the process advances to step S806.

[0051] In step S805, a packet reception unit 302 extracts processing data from the packet and transmits it to the processing data transmission unit 303, which passes that data to the processing unit 304. Then, the process advances to step S807. In step S806, the packet recep-

tion unit 302 instructs a packet modification unit 308 to set a stall flag 404 of the packet to be "1".

[0052] On the other hand, the packet reception unit 302 checks in step S807 if the preset value of the invalidation instruction register 701 is "1". If it is determined that the preset value is "1" ("YES" in step S807), the process advances to step S808. On the other hand, if it is determined that the preset value is not "1" ("NO" in step S807), this processing ends. In step S808, the packet reception unit 302 instructs the packet modification unit 308 to set the data valid flag 401 to be "0".

[0053] As described above, even when the data path branches, since the module includes the invalidation instruction register 701, which is set as needed, a packet can be transmitted to all of a plurality of waiting modules.

Other Embodiments

[0054] Aspects of the present invention can also be realized by a computer of a system or apparatus (or devices such as a CPU or MPU) that reads out and executes a program recorded on a memory device to perform the functions of the above-described embodiment(s), and by a method, the steps of which are performed by a computer of a system or apparatus by, for example, reading out and executing a program recorded on a memory device to perform the functions of the above-described embodiment(s). For this purpose, the program is provided to the computer for example via a network or from a recording medium of various types serving as the memory device (e.g., computer-readable storage medium).

[0055] While the present invention has been described with reference to exemplary embodiments, it is to be understood that the invention is not limited to the disclosed exemplary embodiments. The scope of the following claims is to be accorded the broadest interpretation so as to encompass all such modifications and equivalent structures and functions.

Claims

1. An information processing apparatus including a plurality of modules (201 - 204), which are connected to a ring bus (205), and each of which is configured to receive a packet that stores data from one neighboring module, and to transmit the packet to the other neighboring module after predetermined processing, each module (201 - 204) comprises:

identification means (301) for identifying a packet received from another module to determine whether or not the received packet is a packet to be processed by a self-module;
reception means (302) for, when said identification means (301) determines that the received packet is a packet to be processed by the self-

module, extracting data to be processed from the packet;
 processing means (304) for processing the extracted data; and
 transmission means (307, 309, 310) for transmitting the packet to the other module;
characterized in that each module further comprises:

modification means (308) for, when said processing means (304) is ready to process the data, setting a first flag (401) indicating validity of data in the received packet to have a value indicating that the data is invalid;
 wherein when said processing means (304) is not ready to process the data, said modification means (308) modifies a value of a second flag (404) in the received packet from an initial value.

2. The apparatus according to claim 1, **characterized in that** each module (201 - 204) further comprises a register (701) having a preset value indicating whether or not the self-module is permitted to set the first flag (401), and
 when said processing means (304) is ready to process the data and the preset value of said register (701) has a value which permits to set the first flag (401), said modification means (308) sets the first flag (401) to have the value indicating that the data of the packet is invalid.
3. The apparatus according to claim 1, **characterized in that** said identification means (301) further determines, in association with the received packet, values of the first flag (401) and the second flag (404) and whether or not a transmission source of the packet is the self-module, and
 when said identification means (301) determines that the first flag (401) is set to have a value indicating that the data of the received packet is valid, the second flag (404) is set to have the initial value, and the transmission source of the received packet is the self-module,
 said modification means (308) sets the first flag (401) in the received packet to have the value indicating that the data is invalid, and
 said transmission means (307, 309, 310) transmits, to the other module, the packet in which said modification means (308) sets the first flag (401) in the received packet to have the value indicating that the data of the packet is invalid.
4. The apparatus according to claim 1 or 3, **characterized in that** each module (201 - 204) further comprises generation means (306) for generating a packet including data processed by said processing

means,
 said identification means (301) further determines a value of the first flag (401) in the received packet, and
 when said identification means (301) determines that the first flag (401) is set to have the value indicating that the data in the received packet is invalid, said generation means (306) generates a packet by storing the data processed by said processing means (304) in a data storage field of the received packet,
 setting the first flag (401) to have a value indicating that the data in the packet is valid,
 setting the second flag (404) to have the initial value, and
 setting an identifier (402) used to specify the self-module in a transmission source of the packet to be generated, and
 said transmission means (307, 309, 310) transmits the packet generated by said generation means (306) to the other module.

5. An operation method of an information processing apparatus including a plurality of modules (201 - 204), which are connected to a ring bus (205), and each of which is configured to receive a packet that stores data from one neighboring module, and to transmit the packet to the other neighboring module after predetermined processing,
 each module executing steps of:

identifying a packet received from another module to determine whether or not the received packet is a packet to be processed by a self-module;
 extracting, when it is determined in said identifying step that the received packet is a packet to be processed by the self-module, data to be processed from the packet;
 controlling processing means (304) to process the extracted data; and
 transmitting the packet to the other module;
characterized in that each module further executing steps of:

setting, when the processing means is ready to process the data, a first flag (401) indicating validity of data in the received packet to have a value indicating that the data is invalid; and
 modifying, when the processing means (304) is not ready to process the data, a value of a second flag (404) in the received packet from an initial value.

in each module,

in the setting step of the operation method, when the processing means (304) is ready to process the data and the preset value of the register has a value which permits to set the first flag (401), the first flag (401) is set to have the value indicating that the data of the packet is invalid.

7. The method according to claim 5, **characterized in that** in the identifying step, values of the first flag (401) and the second flag (404) and whether or not a transmission source of the packet is the self-module are further determined in association with the received packet, and when it is determined in the identifying step that the first flag (401) is set to have a value indicating that the data of the received packet is valid, the second flag (404) is set to have the initial value, and the transmission source of the received packet is the self-module, in the setting step, the first flag (401) in the received packet is set to have the value indicating that the data is invalid, and in the transmitting step, the packet in which the first flag (401) in the received packet is set to have the value indicating that the data of the packet is invalid in the modification step is transmitted to the other module.
8. The method according to claim 5 or 7, **characterized in that** each module (201 - 204) further executes a step of generating a packet including data processed by the processing means, in the identifying step, a value of the first flag (401) in the received packet is further determined, and when it is determined in the identifying step that the first flag (401) is set to have the value indicating that the data in the received packet is invalid, in the generating step, a packet is generated by storing the data processed by the processing means in a data storage field of the received packet, setting the first flag (401) to have a value indicating that the data in the packet is valid, setting the second flag (404) to have the initial value, and setting an identifier (402) used to specify the self-module in a transmission source of the packet to be generated, and in the transmitting step, the packet generated in the generation step is transmitted to the other module.
9. A computer program comprising instructions which, when run on a computer, causes that computer to execute an operation method according to any one of claims 5 to 8.
10. A computer readable storage medium storing a computer program according to claim 9.

Patentansprüche

1. Informationsverarbeitungsvorrichtung mit mehreren mit einem Ringbus (205) verbundenen Modulen (201 - 204), von denen ein jedes konfiguriert ist, ein Daten speicherndes Paket von einem benachbarten Modul zu empfangen und das Paket an das andere benachbarte Modul nach vorbestimmter Verarbeitung zu übertragen, wobei jedes Modul (201 - 204) umfasst:

eine Identifizierungseinrichtung (301) zum Identifizieren eines von einem anderen Modul empfangenen Pakets, um zu bestimmen, ob das empfangene Paket ein von einem Eigenmodul zu verarbeitendes Paket ist oder nicht;
eine Empfangseinrichtung (302) zum Extrahieren von zu verarbeitenden Daten aus dem Paket, wenn die Identifizierungseinrichtung (301) bestimmt, dass das empfangene Paket ein vom Eigenmodul zu verarbeitendes Paket ist;
eine Verarbeitungseinrichtung (304) zum Verarbeiten der extrahierten Daten; und
eine Übertragungseinrichtung (307, 309, 310) zum Übertragen des Pakets an das andere Modul;
dadurch gekennzeichnet, dass jedes Modul ferner umfasst:

eine Änderungseinrichtung (308) zum Einstellen, dass ein erstes, Gültigkeit von Daten im empfangenen Paket angegebendes Flag (401) einen die Ungültigkeit der Daten angegebenden Wert aufweist, wenn die Verarbeitungseinrichtung (304) bereit ist, die Daten zu verarbeiten;
wobei die Änderungseinrichtung (308) einen Wert eines zweiten Flags (404) im empfangenen Paket ausgehend vom Anfangswert ändert, wenn die Verarbeitungseinrichtung (304) nicht bereit ist, die Daten zu verarbeiten.

2. Vorrichtung nach Anspruch 1, **dadurch gekennzeichnet, dass** jedes Modul (201 - 204) ferner ein Register (701) mit einem voreingestellten Wert umfasst, der angibt, ob es dem Eigenmodul erlaubt ist, das erste Flag (401) einzustellen oder nicht, und

wenn die Verarbeitungseinrichtung (304) bereit ist, die Daten zu verarbeiten, und der voreingestellte Wert des Registers (701) einen Wert aufweist, der es erlaubt, das erste Flag (401) einzustellen, die Änderungseinrichtung (308) das erste Flag (401) so einstellt, dass es den die Ungültigkeit der Daten des Pakets angegebenden Wert aufweist.

3. Vorrichtung nach Anspruch 1, **dadurch gekennzeichnet, dass** die Identifizierungseinrichtung (301) im Zusammenhang mit dem empfangenen Paket ferner Werte des ersten Flags (401) und des zweiten Flags (404) bestimmt und bestimmt, ob eine Übertragungsquelle des Pakets das Eigenmodul ist oder nicht, und

wenn die Identifizierungseinrichtung (301) bestimmt, dass das erste Flag (401) so eingestellt ist, dass es einen die Gültigkeit der Daten des empfangenen Pakets angehenden Wert aufweist, das zweite Flag (404) so eingestellt wird, dass es den Anfangswert aufweist, und die Übertragungsquelle des empfangenen Pakets das Eigenmodul ist, die Änderungseinrichtung (308) das erste Flag (401) im empfangenen Paket so einstellt, dass es den die Ungültigkeit der Daten angehenden Wert aufweist, und die Übertragungseinrichtung (307, 309, 310) das Paket, in welchem die Änderungseinrichtung (308) das erste Flag (401) im empfangenen Paket so einstellt, dass es den die Ungültigkeit der Daten des Pakets angehenden Wert aufweist, an das andere Modul überträgt.

4. Vorrichtung nach Anspruch 1 oder 3, **dadurch gekennzeichnet, dass** jedes Modul (201 - 204) ferner eine Erzeugungseinrichtung (306) zum Erzeugen eines Pakets mit von der Verarbeitungseinrichtung verarbeiteten Daten umfasst,

die Identifizierungseinrichtung (301) ferner einen Wert des ersten Flags (401) im empfangenen Paket bestimmt, und wenn die Identifizierungseinrichtung (301) bestimmt, dass das erste Flag (401) so eingestellt ist, dass es den die Ungültigkeit der Daten im empfangenen Paket angehenden Wert aufweist, die Erzeugungseinrichtung (306) ein Paket erzeugt durch:

Speichern der von der Verarbeitungseinrichtung (304) verarbeiteten Daten in einem Datenspeicherfeld des empfangenen Pakets, Einstellen des ersten Flags (401), sodass es einen die Gültigkeit der Daten im Paket angehenden Wert aufweist, Einstellen des zweiten Flags (404), sodass es den Anfangswert aufweist, und Einstellen eines für die Spezifizierung des Eigenmoduls in einer Übertragungsquelle des zu erzeugenden Pakets verwendeten Identifizierers (402), und dass die Übertragungseinrichtung (307,

309, 310) das von der Erzeugungseinrichtung (306) erzeugte Paket an das andere Modul überträgt.

5. Betriebsverfahren einer Informationsverarbeitungsvorrichtung mit mehreren mit einem Ringbus (205) verbundenen Modulen (201 - 204), von denen ein jedes konfiguriert ist, ein Daten speicherndes Paket von einem benachbarten Modul zu empfangen und das Paket an das andere benachbarte Modul nach vorbestimmter Verarbeitung zu übertragen,

wobei jedes Modul folgende Schritte durchführt:

Identifizieren eines von einem anderen Modul empfangenen Pakets, um zu bestimmen, ob das empfangene Paket ein von einem Eigenmodul zu verarbeitendes Paket ist oder nicht; Extrahieren von zu verarbeitenden Daten aus dem Paket, wenn im Identifizierungsschritt bestimmt wird, dass das empfangene Paket ein vom Eigenmodul zu verarbeitendes Paket ist; Steuern einer Verarbeitungseinrichtung (304), die extrahierten Daten zu verarbeiten; und Übertragen des Pakets an das andere Modul;

dadurch gekennzeichnet, dass jedes Modul ferner folgende Schritte durchführt:

Einstellen, dass ein erstes, die Gültigkeit von Daten im empfangenen Paket angehenes Flag (401) einen die Ungültigkeit der Daten angehenden Wert aufweist, wenn die Verarbeitungseinrichtung bereit ist, die Daten zu verarbeiten; und Ändern eines Anfangswertes eines zweiten Flags (404) im empfangenen Paket in einen Wert, wenn die Verarbeitungseinrichtung (304) nicht bereit ist, die Daten zu verarbeiten.

6. Verfahren nach Anspruch 5, **dadurch gekennzeichnet, dass** jedes Modul ferner ein Register (701) mit einem voreingestellten Wert umfasst, der angibt, ob es dem Eigenmodul erlaubt ist, das erste Flag (401) einzustellen oder nicht, und

in jedem Modul im Einstellschritt des Betriebsverfahrens das erste Flag (401) so eingestellt wird, dass es den die Ungültigkeit der Daten des Pakets angehenden Wert aufweist, wenn die Verarbeitungseinrichtung (304) bereit ist, die Daten zu verarbeiten, und der voreingestellte Wert des Registers

einen Wert aufweist, der es erlaubt, das erste Flag (401) einzustellen.

7. Verfahren nach Anspruch 5, **dadurch gekennzeichnet, dass** im Identifizierungsschritt im Zusammenhang mit dem empfangenen Paket ferner Werte des ersten Flags (401) und des zweiten Flags (404) bestimmt werden und bestimmt wird, ob eine Übertragungsquelle des Pakets das Eigenmodul ist oder nicht, und

wenn im Identifizierungsschritt bestimmt wird, dass das erste Flag (401) so eingestellt ist, dass es einen die Gültigkeit der Daten des empfangenen Pakets angehenden Wert aufweist, das zweite Flag (404) so eingestellt wird, dass es den Anfangswert aufweist, und die Übertragungsquelle des empfangenen Pakets das Eigenmodul ist, im Einstellschritt das erste Flag (401) im empfangenen Paket so eingestellt wird, dass es den die Ungültigkeit der Daten angehenden Wert aufweist, und im Übertragungsschritt das Paket, in welchem im Änderungsschritt das erste Flag (401) im empfangenen Paket so eingestellt wird, dass es den die Ungültigkeit der Daten des Pakets angehenden Wert aufweist, an das andere Modul übertragen wird.

8. Verfahren nach Anspruch 5 oder 7, **dadurch gekennzeichnet, dass** jedes Modul (201 - 204) ferner einen Schritt des Erzeugens eines Pakets mit von der Verarbeitungseinrichtung verarbeiteten Daten durchführt,

im Identifizierungsschritt ferner ein Wert des ersten Flags (401) im empfangenen Paket bestimmt wird, und wenn im Identifizierungsschritt bestimmt wird, dass das erste Flag (401) so eingestellt ist, dass es den die Ungültigkeit der Daten im empfangenen Paket angehenden Wert aufweist, dass im Erzeugungsschritt ein Paket erzeugt wird durch:

Speichern der von der Verarbeitungseinrichtung verarbeiteten Daten in einem Datenspeicherfeld des empfangenen Pakets, Einstellen des ersten Flags (401), sodass es einen die Gültigkeit der Daten im Paket angehenden Wert aufweist, Einstellen des zweiten Flags (404), sodass es den Anfangswert aufweist, und Einstellen eines für die Spezifizierung des Eigenmoduls in einer Übertragungsquelle des zu erzeugenden Pakets verwendeten Identifizierers (402), und

dass im Übertragungsschritt das im Erzeugungsschritt erzeugte Paket an das andere Modul übertragen wird.

9. Computerprogramm mit Befehlen, welches bei Ausführung auf einem Computer diesen veranlasst, ein Betriebsverfahren nach einem der Ansprüche 5 bis 8 durchzuführen.
10. Computerlesbares Speichermedium, das ein Computerprogramm nach Anspruch 9 speichert.

Revendications

1. Appareil de traitement d'information comprenant une pluralité de modules (201 à 204), qui sont connectés à un bus en anneau (205), chacun d'entre eux étant configuré pour recevoir d'un module voisin un paquet qui transporte des données et pour transmettre le paquet vers l'autre module voisin après un traitement prédéterminé, chacun des modules (201 à 204) comprenant :

un moyen d'identification (301) destiné à identifier un paquet reçu d'un autre module afin de déterminer si le paquet reçu est ou non un paquet qui doit être traité par le module lui-même ;
un moyen de réception (302) qui, lorsque ledit moyen d'identification (301) détermine que le paquet reçu est un paquet qui doit être traité par le module lui-même, est destiné à extraire du paquet des données à traiter ;
un moyen de traitement (304) destiné à traiter les données extraites ;
un moyen de transmission (307, 309, 310) destiné à transmettre le paquet vers l'autre module ;
caractérisé en ce que chaque module comprend en outre :

un moyen de modification (308) qui, lorsque ledit moyen de traitement (304) est prêt à traiter les données, est destiné à armer un premier drapeau (401) indiquant la validité des données dans le paquet reçu pour prendre une valeur indiquant que les données sont invalides ;
dans lequel, lorsque le moyen de traitement (304) n'est pas prêt à traiter les données, ledit moyen de modification (308) modifie une valeur d'un second drapeau (404) dans le paquet reçu par rapport à une valeur initiale.

2. Appareil selon la revendication 1, **caractérisé en ce que** chaque module (201 à 204) comprend en outre un registre (701) ayant une valeur prédéterminée indiquant si le module lui-même est ou non autorisé

à armer le premier drapeau (401) ;
dans lequel, lorsque le moyen de traitement (304)
est prêt à traiter les données et la valeur prédéter-
minée dudit registre (701) a une valeur qui autorise
l'armement du premier drapeau (401), ledit moyen
de modification (308) arme le premier drapeau (401)
pour lui donner une valeur indiquant que les données
du paquet sont invalides.

3. Appareil selon la revendication 1, **caractérisé en ce que** ledit moyen d'identification (301) détermine en outre, en association avec le paquet reçu, des valeurs du premier drapeau (401) et du second drapeau (404) et si une source de transmission du paquet se situe ou non dans le module lui-même ; et lorsque le moyen d'identification (301) détermine que le premier drapeau (401) est armé à une valeur indiquant que les données du paquet reçu sont valides, le second drapeau (404) est armé à la valeur initiale et la source de transmission du paquet reçu est le module lui-même ;
ledit moyen de modification (308) arme le premier drapeau (401) dans le paquet reçu à la valeur indiquant que les données sont invalides ; et
ledit moyen de transmission (307, 309, 310) transmet vers l'autre module le paquet dans lequel ledit moyen de modification (308) a armé le premier drapeau (401) dans le paquet reçu à la valeur indiquant que les données du paquet sont invalides.
4. Appareil selon la revendication 1 ou 3, **caractérisé en ce que** chaque module (201 à 204) comprend en outre un moyen de génération (306) destiné à générer un paquet incluant des données traitées par ledit moyen de traitement ;
ledit moyen d'identification (301) détermine en outre une valeur du premier drapeau (401) dans le paquet reçu ; et
lorsque ledit moyen d'identification (301) détermine que le premier drapeau (401) est armé à la valeur indiquant que les données dans le paquet reçu sont invalides,
ledit moyen de génération (306) génère un paquet :

 en stockant les données traitées par ledit moyen de traitement (304) dans un champ de stockage de données du paquet reçu ;
 en armant le premier drapeau (401) à une valeur indiquant que les données dans le paquet sont valides ;
 en armant le second drapeau (404) à la valeur initiale ; et
 en armant un identificateur (402) utilisé pour spécifier le module lui-même dans une source de transmission du paquet à produire ; et

ledit moyen de transmission (307, 309, 310) transmet le paquet généré par ledit moyen de génération

(306) vers l'autre module.

5. Procédé de fonctionnement d'un appareil de traitement d'information comprenant une pluralité de modules (201 à 204), qui sont connectés à un bus en anneau (205), chacun d'entre eux étant configuré pour recevoir d'un module voisin un paquet qui transporte des données et pour transmettre le paquet vers l'autre module voisin après un traitement prédéterminé, chacun des modules exécutant des étapes consistant à :

 identifier un paquet reçu d'un autre module afin de déterminer si le paquet reçu est ou non un paquet qui doit être traité par le module lui-même ;
 extraire du paquet des données à traiter, lorsque ladite étape d'identification détermine que le paquet reçu est un paquet qui doit être traité par le module lui-même ;
 commander un moyen de traitement (304) pour traiter les données extraites ; et
 transmettre le paquet vers l'autre module ;
caractérisé en ce que chaque module exécute en outre les étapes consistant à :

 lorsque le moyen de traitement est prêt à traiter les données, armer un premier drapeau (401) indiquant la validité des données dans le paquet reçu pour prendre une valeur indiquant que les données sont invalides ; et
 lorsque le moyen de traitement (304) n'est pas prêt à traiter les données, modifier une valeur d'un second drapeau (404) dans le paquet reçu par rapport à une valeur initiale.
6. Procédé selon la revendication 5, **caractérisé en ce que** chaque module comprend en outre un registre (701) ayant une valeur prédéterminée indiquant si le module lui-même est ou non autorisé à armer le premier drapeau (401) ; et
dans chaque module,
dans l'étape d'armement du procédé de fonctionnement, lorsque le moyen de traitement (304) est prêt à traiter les données et la valeur prédéterminée dudit registre (701) a une valeur qui autorise l'armement du premier drapeau (401), le premier drapeau (401) est armé pour lui donner une valeur indiquant que les données du paquet sont invalides.
7. Procédé selon la revendication 5, **caractérisé en ce que**, dans l'étape d'identification, des valeurs du premier drapeau (401) et du second drapeau (404), de même que le fait qu'une source de transmission du paquet soit le module lui-même, sont déterminés en outre en association avec le paquet reçu ; et lorsqu'il est déterminé dans l'étape d'identification

que le premier drapeau (401) est armé à une valeur indiquant que les données du paquet reçu sont valides, le second drapeau (404) est armé à la valeur initiale et la source de transmission du paquet reçu est le module lui-même ; 5
 dans l'étape d'armement, le premier drapeau (401) du paquet reçu est armé à la valeur indiquant que les données sont invalides ; et
 dans l'étape de transmission, le paquet, dans lequel le premier drapeau (401) dans le paquet reçu est armé à la valeur indiquant que les données du paquet sont invalides au cours de l'étape de modification, est transmis vers l'autre module. 10

8. Procédé selon la revendication 5 ou 7, **caractérisé en ce que** chaque module (201 à 204) comprend en outre une étape de génération d'un paquet incluant des données traitées par le moyen de traitement ; dans l'étape d'identification, une valeur du premier drapeau (401) dans le paquet reçu est en outre déterminée ; et 15
 lorsqu'il est déterminé dans l'étape d'identification que le premier drapeau (401) est armé à la valeur indiquant que les données dans le paquet reçu sont invalides, 20
 au cours de l'étape de génération un paquet est généré : 25

en stockant les données traitées par le moyen de traitement dans un champ de stockage de données du paquet reçu ; 30
 en armant le premier drapeau (401) à une valeur indiquant que les données dans le paquet sont valides ;
 en armant le second drapeau (404) à la valeur initiale ; et 35
 en armant un identificateur (402) utilisé pour spécifier le module lui-même dans une source de transmission du paquet à produire ; et 40

dans l'étape de transmission, le paquet généré dans l'étape de génération est transmis vers l'autre module.

9. Programme d'ordinateur comprenant des instructions qui, lorsqu'elles sont exécutées sur un ordinateur, amènent l'ordinateur à exécuter un procédé de fonctionnement selon l'une quelconque des revendications 5 à 8. 45
 50
10. Support d'enregistrement lisible par ordinateur, portant un programme d'ordinateur selon la revendication 9. 55

FIG. 1

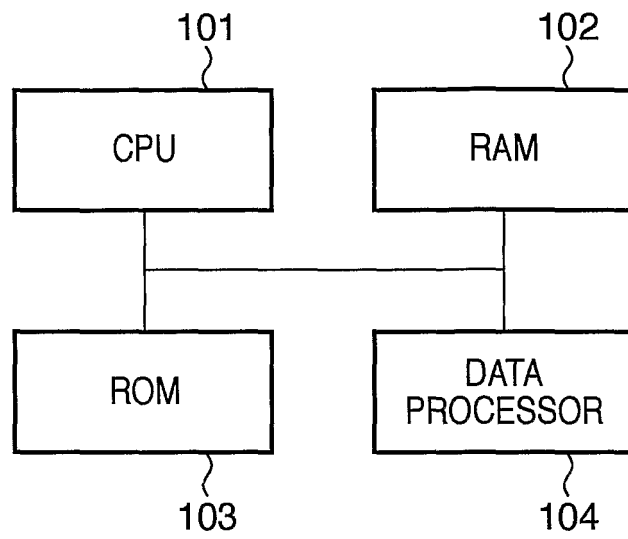
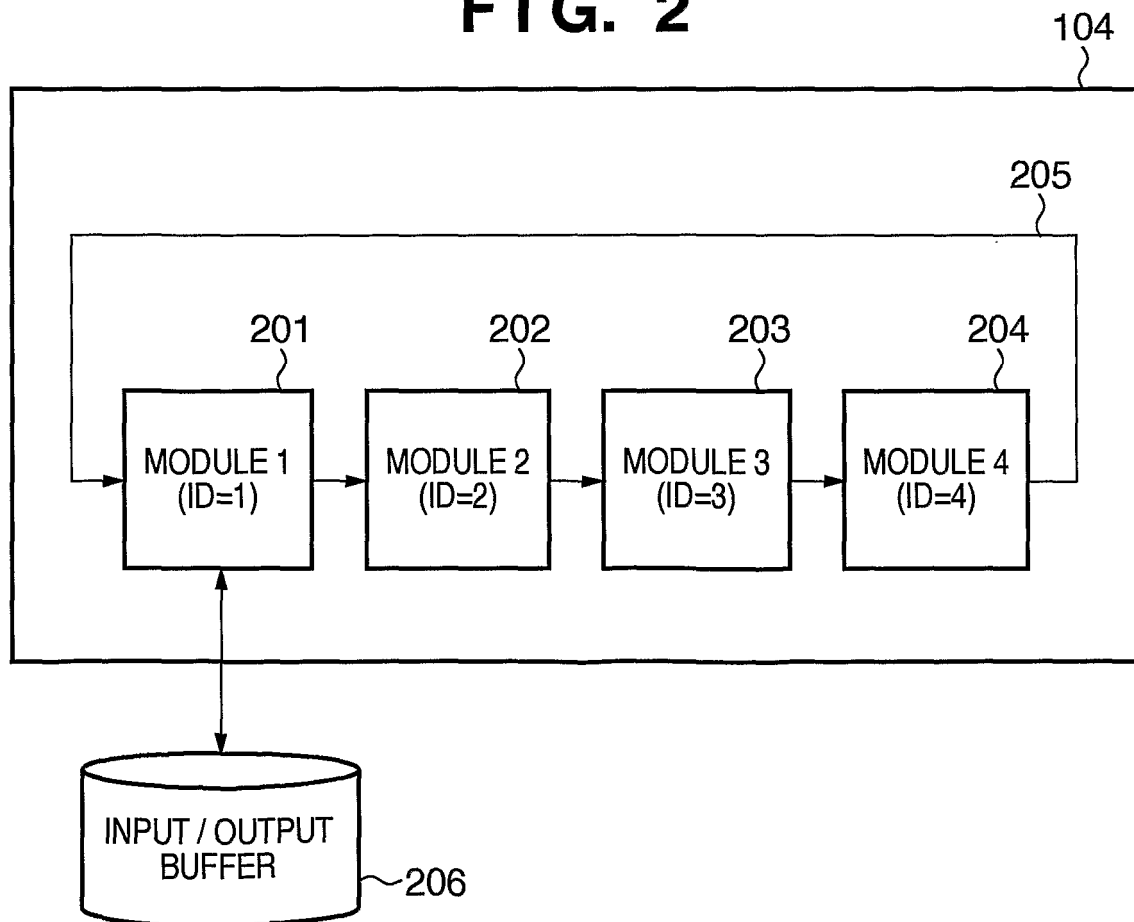


FIG. 2



201, 202, 203, 204

FIG. 3

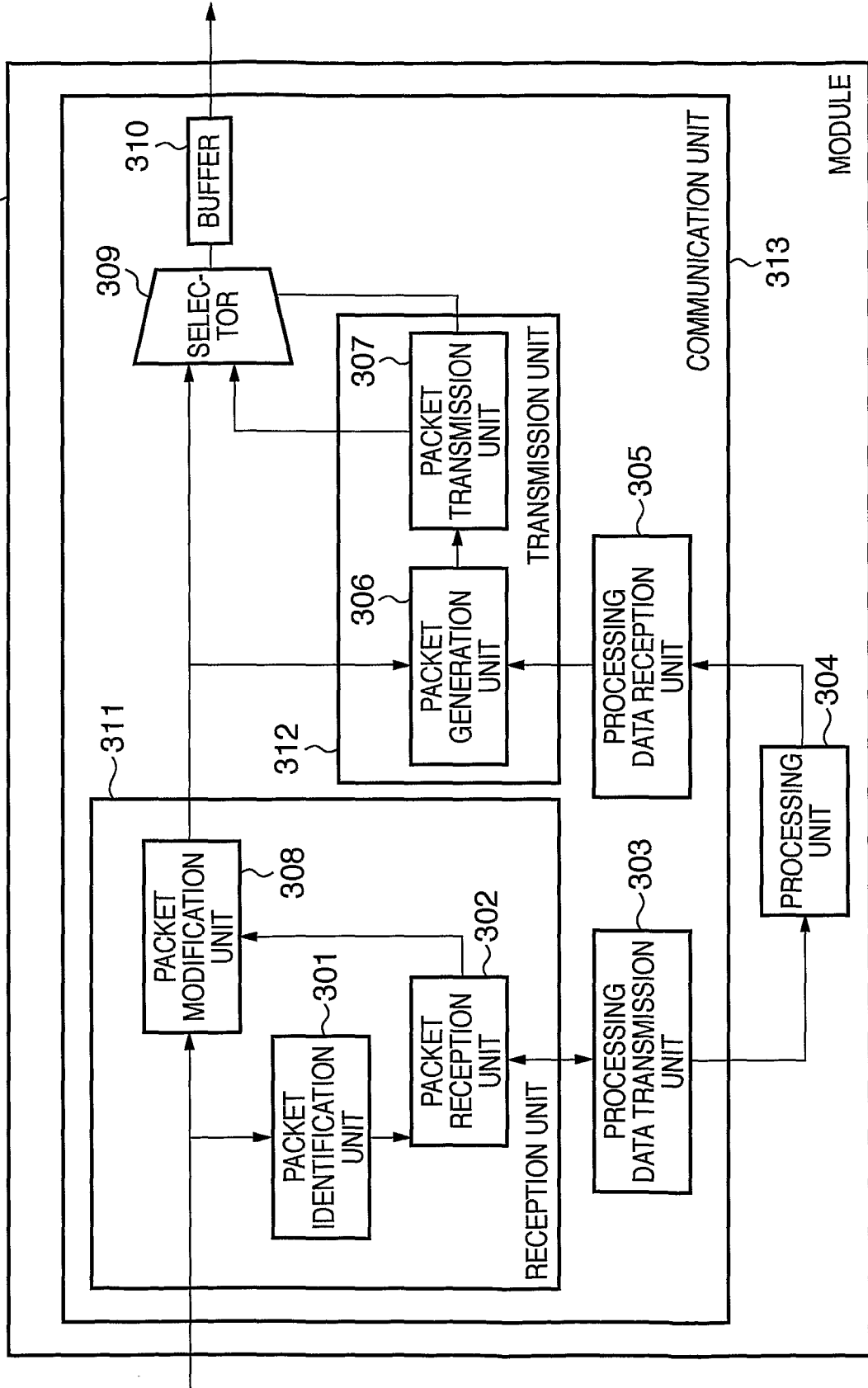


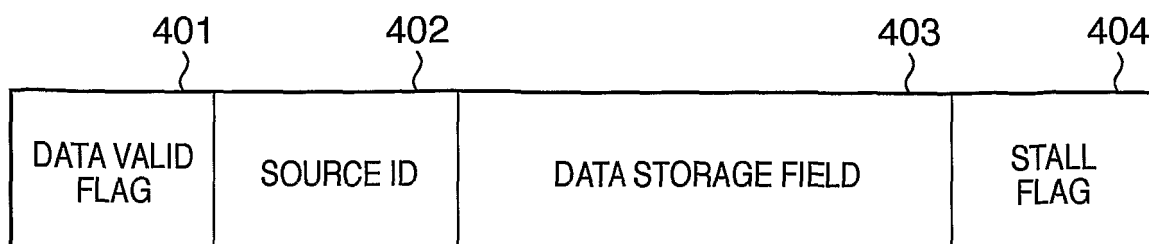
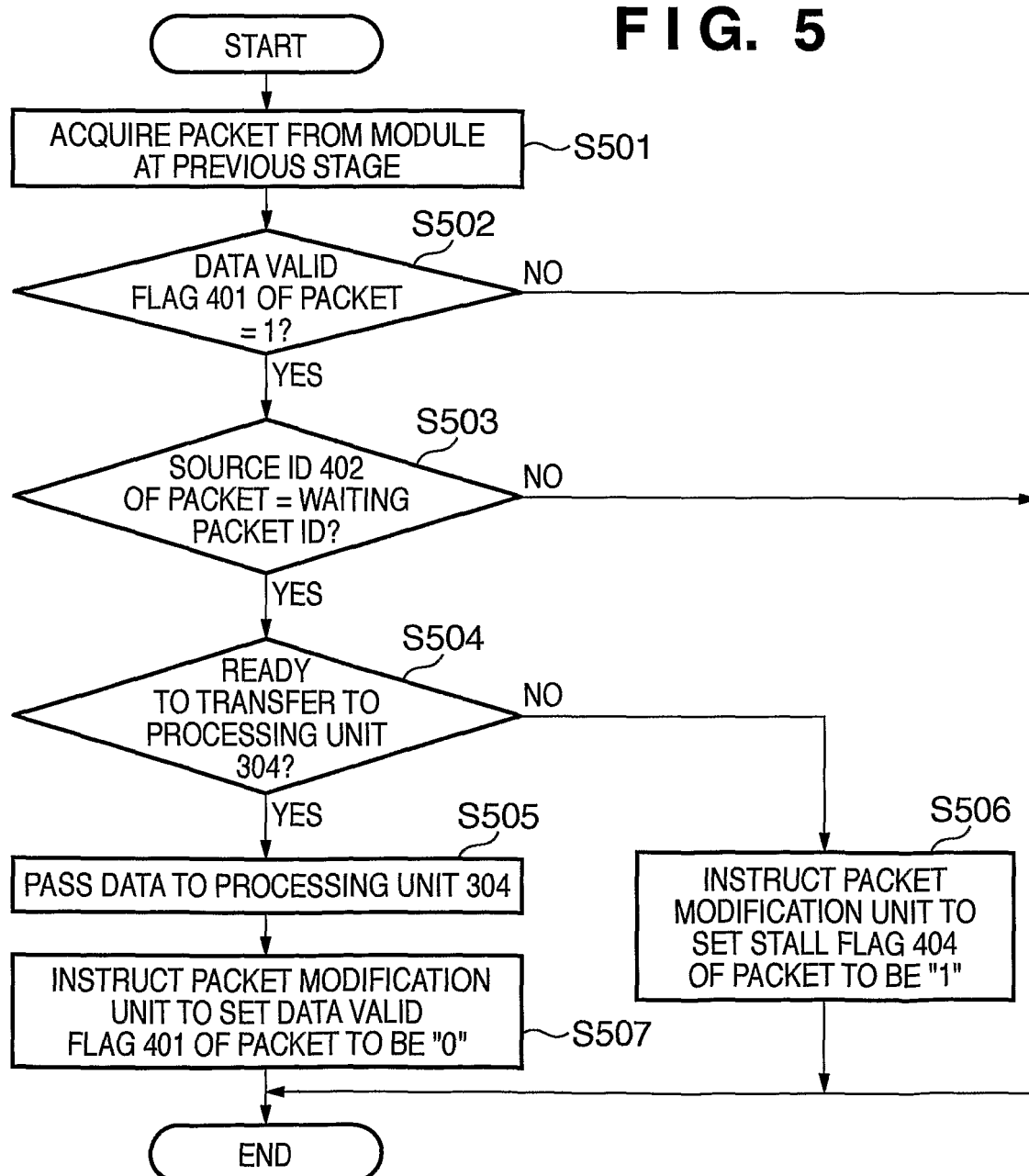
FIG. 4**FIG. 5**

FIG. 6

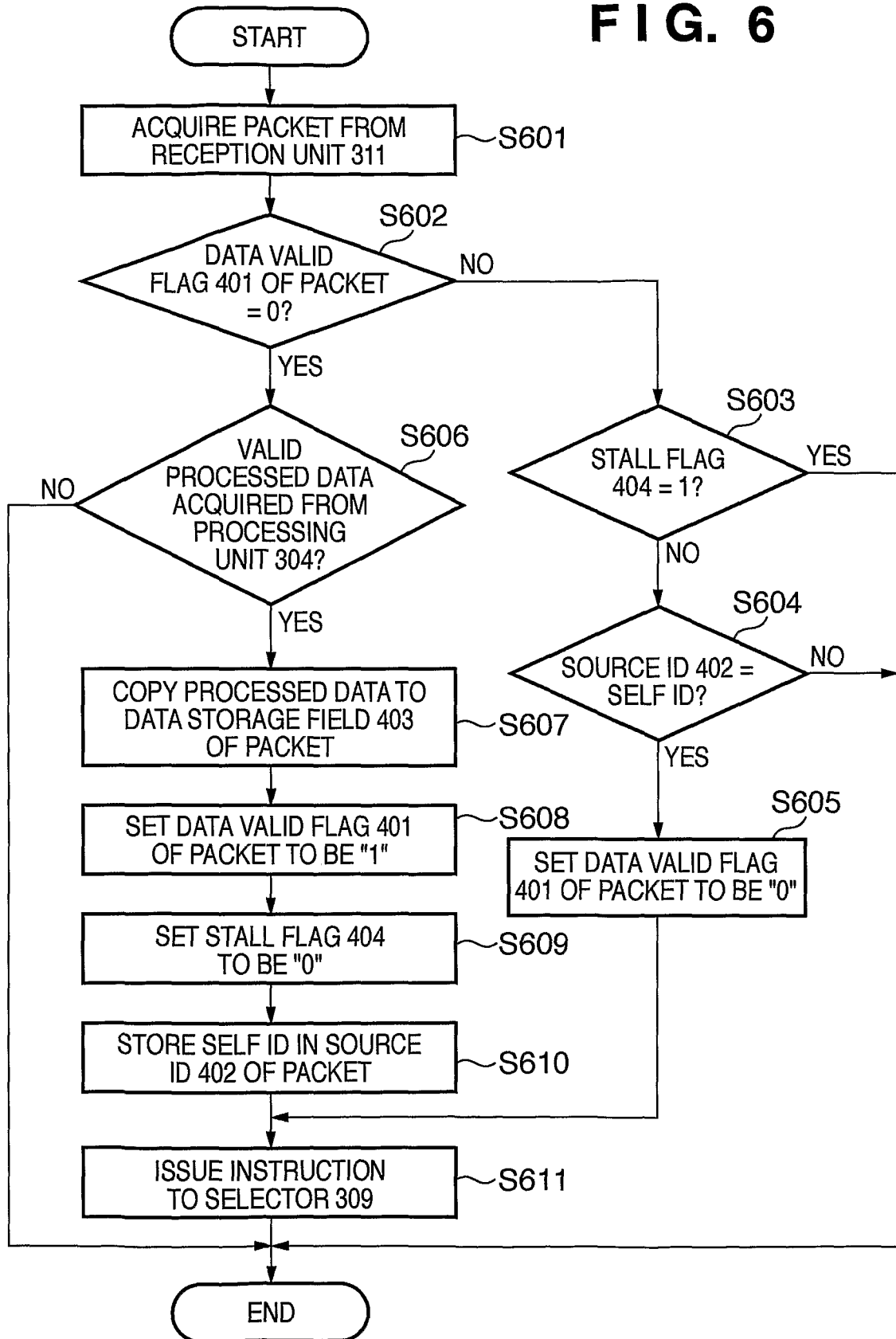


FIG. 7

201, 202, 203, 204

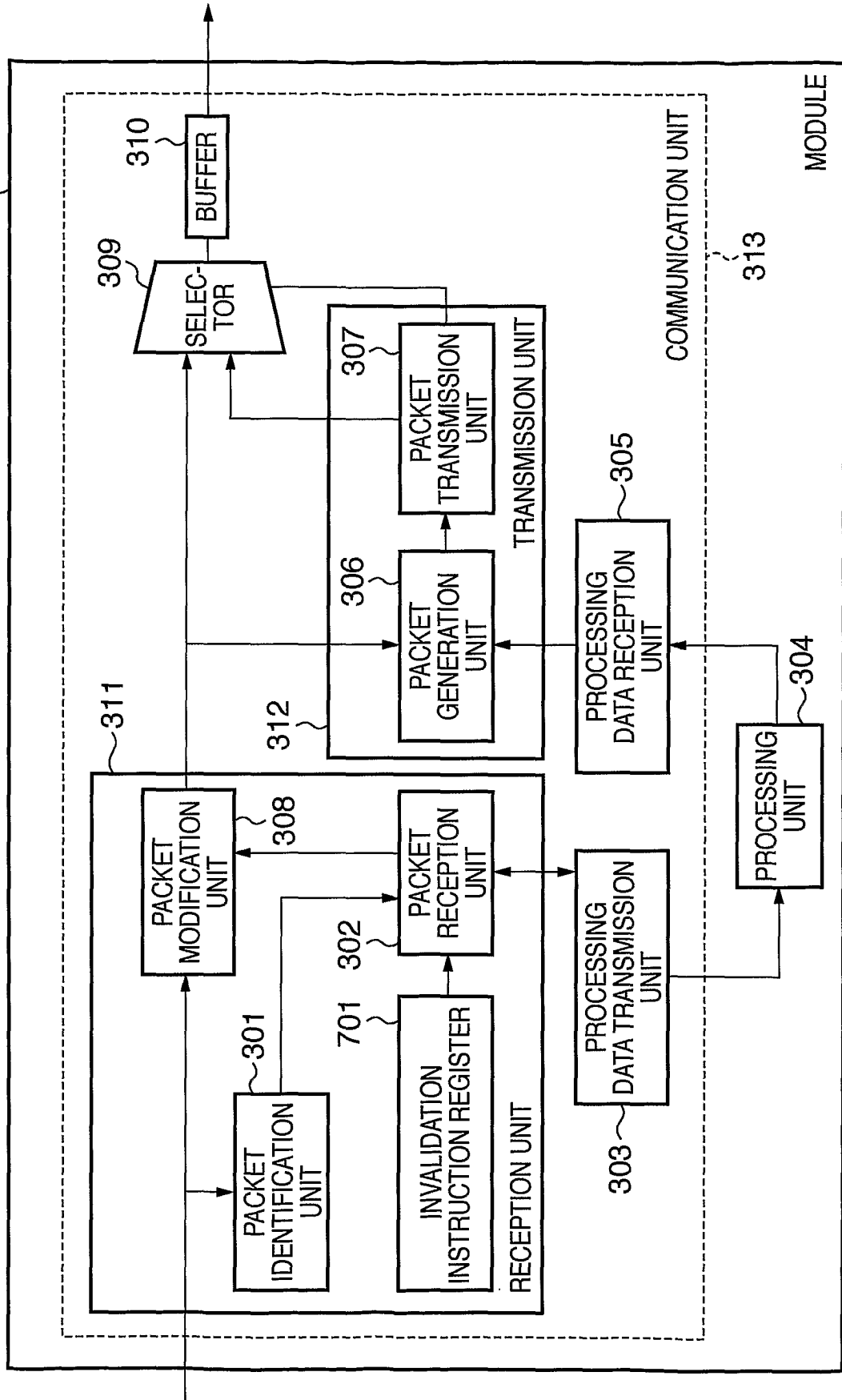
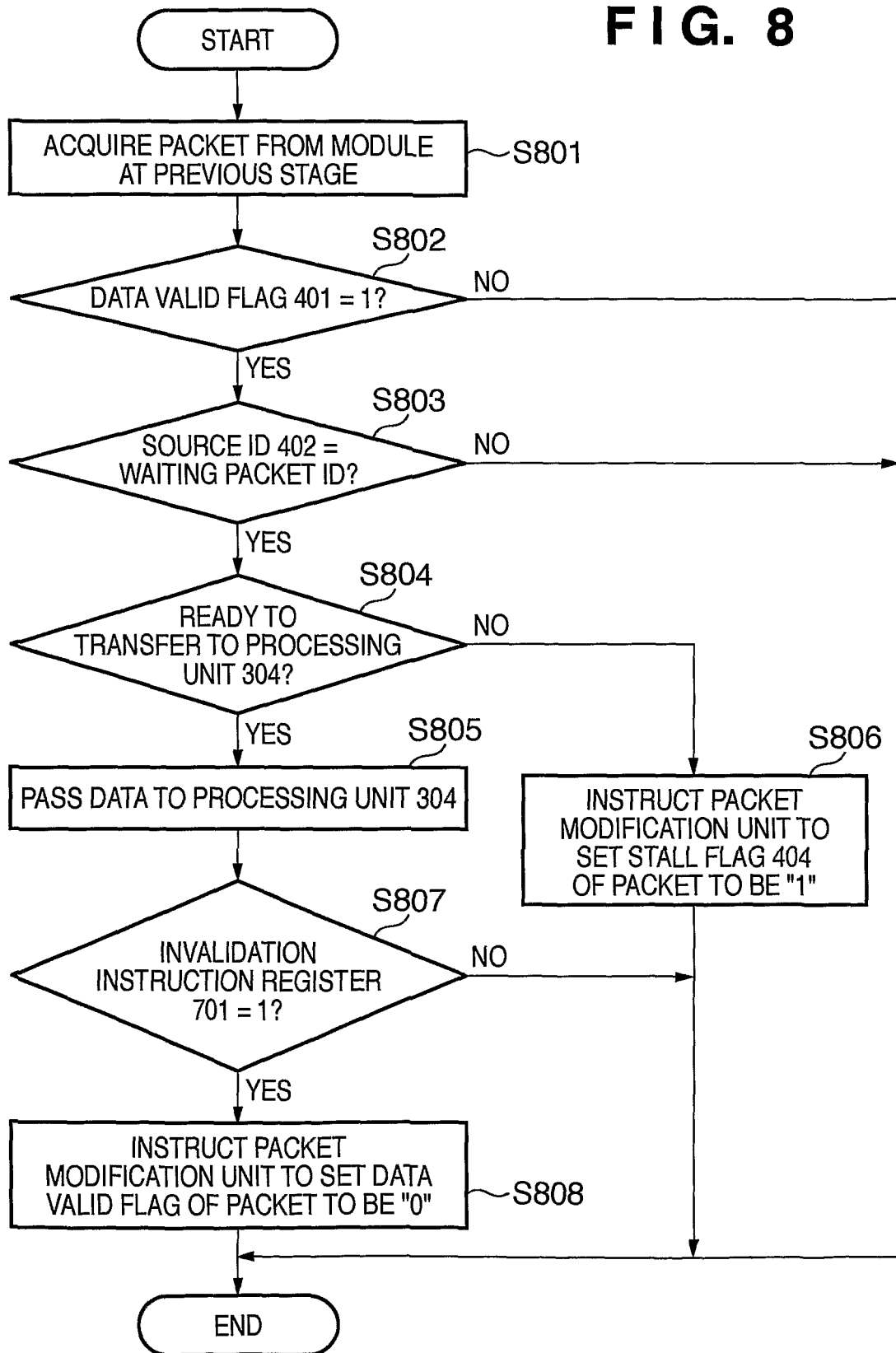


FIG. 8



REFERENCES CITED IN THE DESCRIPTION

This list of references cited by the applicant is for the reader's convenience only. It does not form part of the European patent document. Even though great care has been taken in compiling the references, errors or omissions cannot be excluded and the EPO disclaims all liability in this regard.

Patent documents cited in the description

- US 20020191601 A1 [0005]